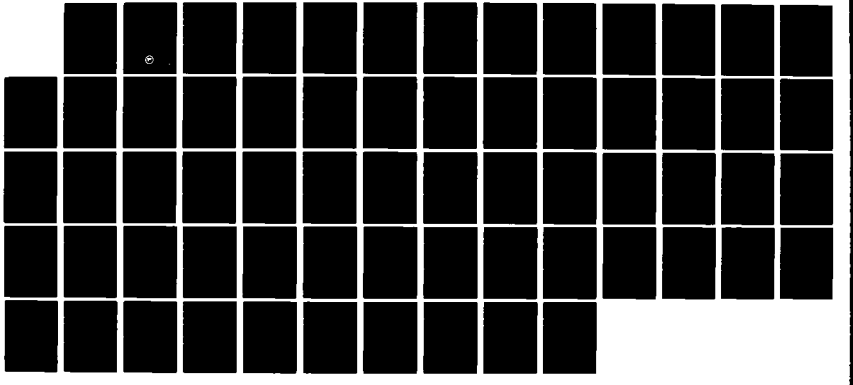
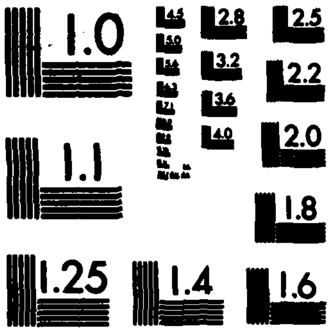
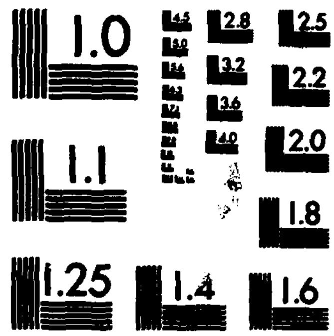
AD-A120 698 A MATHEMATICAL MODEL FOR THE VERIFICATION OF SYSTOLIC NETWORKS(U) PITTSBURGH UNIV PA INST FOR COMPUTATIONAL MATHEMATICS AND APP.. R G MELHEM ET AL. OCT 82 1/1

UNCLASSIFIED ICMA-82-47 N00014-80-C-0455 F/G 12/1 NL

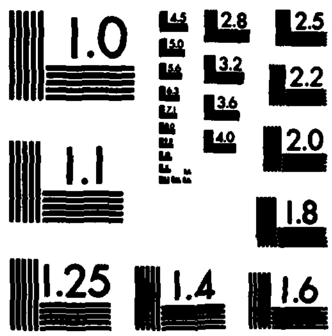




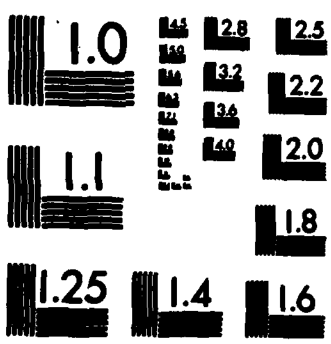
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



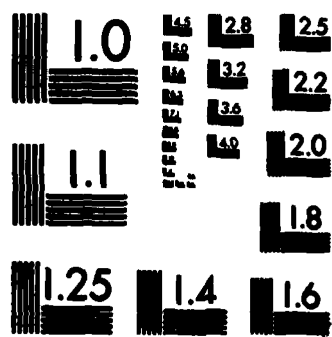
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

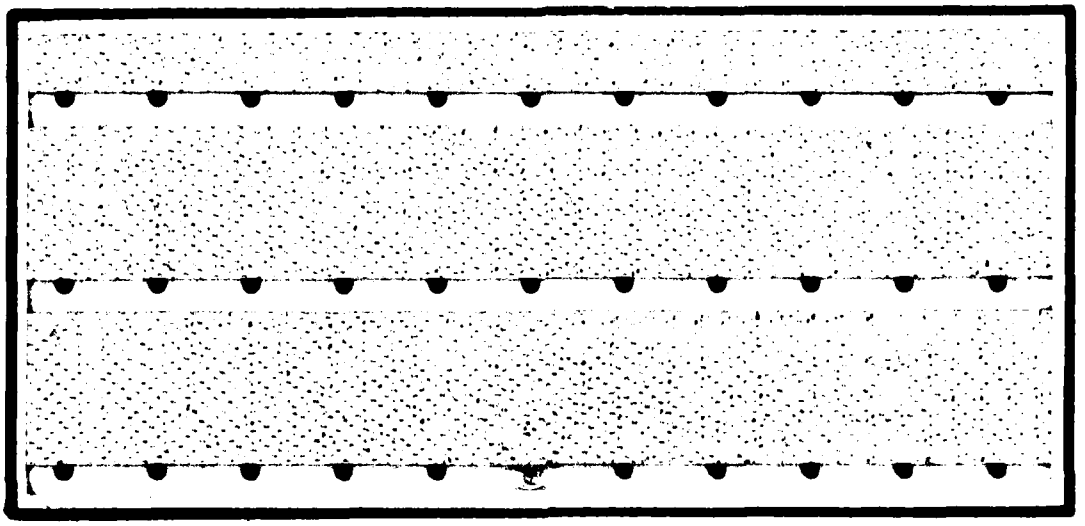


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



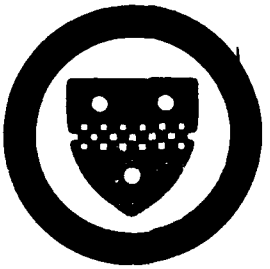
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INSTITUTE FOR COMPUTATIONAL MATHEMATICS AND APPLICATIONS

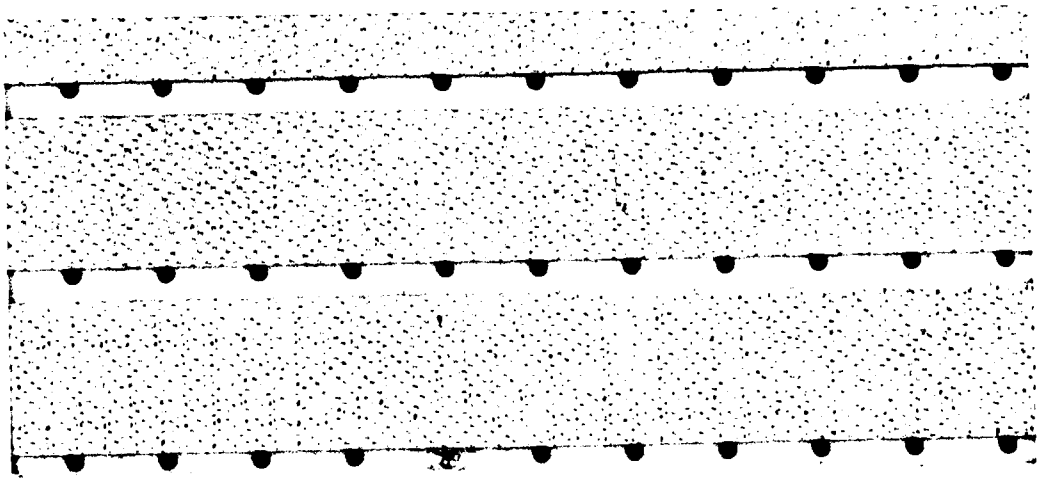


Department of Mathematics and Statistics

University of Pittsburgh



82 10 25 021



CORRECTIONS

p. 20, line +13: ...directed to the right...

p. 39, Figure (9): The arrow for b_0 should be directed upward.

p. 43, line +1: ...of the '$\oplus$' operator...

p. 45, line +8: colored p and s, respectively.

Technical Report ICMA-82-47

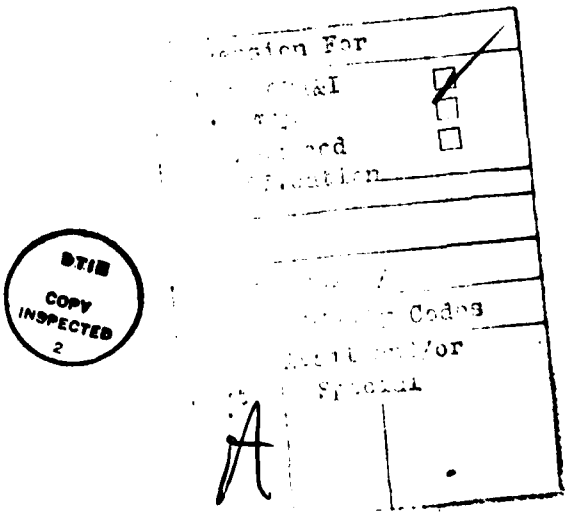
A MATHEMATICAL MODEL FOR THE VERIFICATION OF SYSTOLIC NETWORKS[1)]

Rami G. Melhem
Department of Computer Science
and
Department of Mathematics and Statistics
and
Werner C. Rheinboldt
Department of Mathematics and Statistics

1) This work was supported in part by the Office of Naval Research under Contract N00014-80-C-0455 and the U.S. Air Force Office of Scientific Research under Grant 80-0176.

ABSTRACT

A mathematical model for systolic architectures is suggested and used to verify the operation of certain systolic networks. The data items appearing on the communication links of such a network at successive time units are represented by data sequences and the computations performed by the network-cells are modeled by a system of difference equations involving operations on the various data sequences. The input/output descriptions, which describe the global effect of the computations performed by the network, are obtained by solving this system of difference equations. This input/output description can then be used to verify the operation of the network. The suggested verification technique is applied to four different systolic networks proposed in the literature.



1. Introduction.

Systolic architectures, pioneered by H. T. Kung, are becoming increasingly attractive due to continuous advances in VLSI technology. This type of network architectures has two properties very desirable in VLSI implementations; namely, regularity and the local nature of the interconnections.

A systolic network can be viewed as a network composed of a few types of computational cells, regularly interconnected via local data links and organized such that streams of data flow smoothly within the network. For an introduction to systolic architectures, we refer to [10] where further references to specific examples are given.

As an introductory example, we briefly review a simple systolic network for the computation of one dimensional convolution expressions [10]. More specifically, given a sequence of numbers $\{x_1, x_2, \ldots x_n\}$, and a sequence of weights $\{w_1, w_2, \ldots w_k\}$, we want to compute the sequence $\{y_1, y_2, \ldots y_{n+1-k}\}$ where each y_i is defined by:

$$y_i = \sum_{j=1}^{k} w_j \, x_{i+j-1} \cdot \qquad (1.1)$$

Figure 1 shows the building cell of the 1-D convolution network under discussion. It is a multiply/add cell with a one word memory to store a real number w.

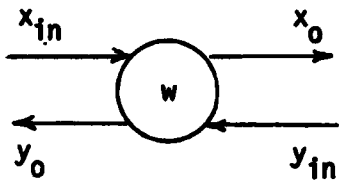


Figure (1)

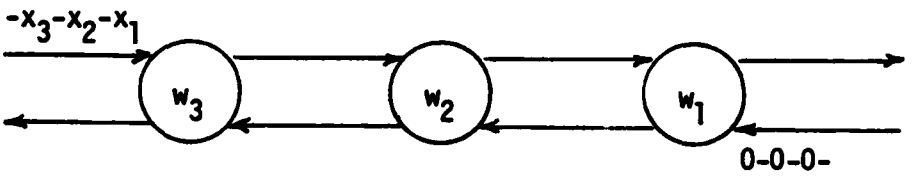
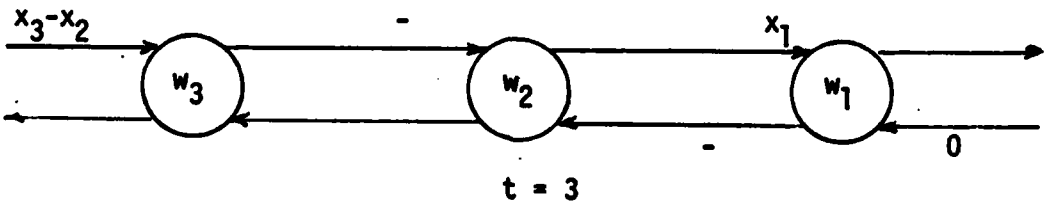
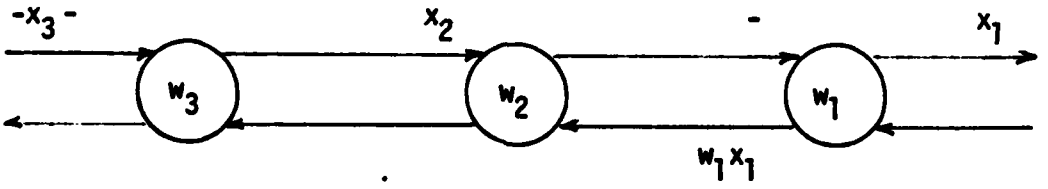


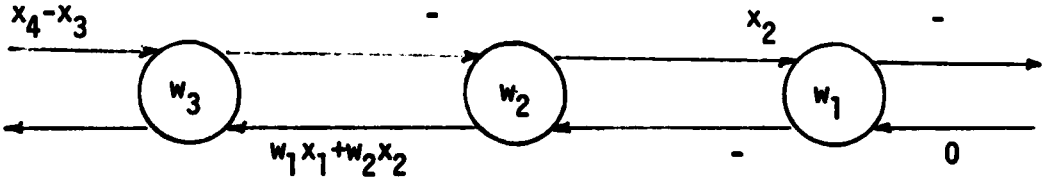
Figure (2)



t = 3



t = 4



t = 5

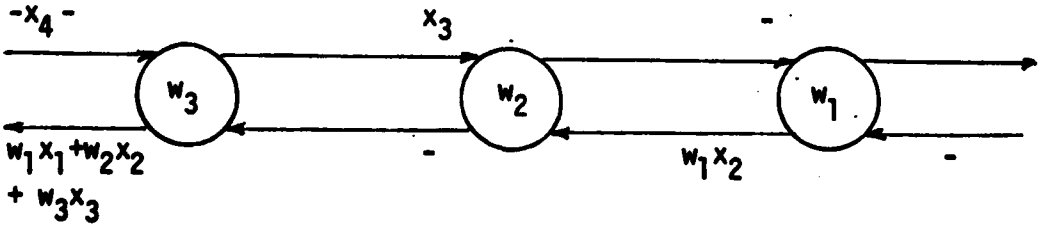


Figure (3)

At each clock pulse, the cell receives two input data items; x_{in} and y_{in}, performs its computation and delivers at the next clock pulse the outputs $x_o = x_{in}$ and $y_o = y_{in} + w\, x_{in}$. Figure 2 shows three such cells connected into a network that performs the convolution calculation for the case k=3. The elements $x_1, x_2, \ldots x_n$ are pumped in at the left end of the network, each separated from the other by one time unit, and zeroes are pumped in at the right end. To illustrate the operation of the array, we show in figure 3 the relative location and value of each data item at times t=3,4,5 and 6, where t=1 is the time at which the array started its execution. By following the data paths, we can convince ourselves that the output of the array will include the sequence $\{y_1, y_2, \ldots y_{n+1-k}\}$.

Although the concept of systolic networks is very well developed, the notation used to describe the input and output data of a systolic network is sometimes ambiguous and reflects poorly the relative timing of the different data streams. Moreover, no rigorous techniques appear to be known for a formal verification of the operation of such networks. To the knowledge of the authors, there has been only one attempt [6] to verify formally the operation of systolic networks based on a proof technique used in the verification of distributed systems [4]. This technique does not make use of the special properties of systolic networks and hence gives only rather general results.

In this paper, we suggest a technique designed specifically for verifying the operation of systolic networks. In section 2.1 the data sequences are introduced to represent the data appearing on the communication links at successive time intervals. In the same section, we discuss the causal operators which model the computations performed by a cell of the network. This concept was primarily inspired by corresponding approaches in systems theory [7].

In section 2.2 and 2.3, we present the mathematical model on which the verification technique is based. This model carries some of the properties of a model called "automaton networks" [3] which in turn is a modification of the von Neumann cellular array [5,11]. However, the two models have more differences than similarities, and are used in completely different contexts.

In section 3 we describe the different steps of the suggested technique and give a simple illustrative example. Finally, in sections 4,5 and 6, we demonstrate the technique by applying it to the verification of some realistic systolic networks that have appeared in the literature.

2. An abstract systolic model.

2.1. Data sequences and causal relations.

We define a data sequence to be an infinite sequence whose elements are members of the set $R_\delta = R \cup \{\delta\}$, where R is the set of real numbers and δ denotes a special element, not belonging to R, called the "don't care element". We extend each one of the four basic arithmetic operations "op" defined on R to R_δ by adding the rule that the result of any such extended arithmetic operation on R_δ involving δ shall equal δ. That is if 'op' = '+', '-', '*' or '÷', then

$$\delta \text{ 'op' } x = x \text{ 'op' } \delta = \delta \quad \text{for all } x \in R_\delta$$

Clearly, operators may also be defined directly on R_δ. For example, we will consider later the binary operator $\oplus$ such that for any $x,y \in R_\delta$,

$$x \oplus y = x + y, \quad \text{if } x,y \neq \delta; \quad x \oplus \delta = \delta \oplus x = x \qquad (2.1)$$

Two other operators that will be used in section 6 are the operators $\min_\delta$ and $\max_\delta$ defined on an ordered pair (x,y), $x,y \in R_\delta$ by

$$\min_\delta(x,y) = \begin{cases} \min\{x,y\} & \text{if } x,y \neq \delta \\ y & \text{if } x=\delta \text{ or } y=\delta \end{cases}$$

and

$$\max_{\delta}(x,y) = \begin{cases} \max\{x,y\} & \text{if } x,y \neq \delta \\ x & \text{if } x=\delta \text{ or } y=\delta. \end{cases}$$

where min{} and max{} carry the usual meaning on R.

Let N be the set of positive integers, then any data sequence η is defined as a mapping from N to R_δ; that is, the image element $\eta(i)$, $i \in N$, is the i^{th} element in the sequence. The set of all data sequences, that is the set of all such mappings, will be denoted by $R_\delta^* = \{ \eta \mid \eta : N \to R_\delta \}$.

Any arithmetic operation on R_δ is extended to R_δ^* by applying the operation element-wise to the elements of the sequences with δ being the result of any undefined operation. For example, if 'op' is a binary operation defined on R_δ, then for all $\eta_1, \eta_2 \in R_\delta^*$, we have η_1 'op' $\eta_2 = \eta_3$ where for all $i \in N$, $\eta_3(i)$ is given by

$$\eta_3(i) = \begin{cases} \eta_1(i) \text{ 'op' } \eta_2(i) & \text{if } \eta_3(i) \text{ is defined} \\ \delta & \text{otherwise.} \end{cases}$$

We will also use scalar operations on sequences. For example, the scalar product of a sequence $\eta \in R_\delta$ and a number $w \in R$ is defined as the sequence $\zeta = w \cdot \eta \in R_\delta^*$ for which

$\zeta(i) = w\ \eta(i)$, $i \epsilon N$.

Given the previous definition of data sequences, we define the set of bounded data sequences $\bar{R}_\delta \subset R_\delta^*$ to contain those sequences having only a finite number of non-δ elements. It is then natural to introduce the termination function $T:\bar{R}_\delta \rightarrow N$ such that for any $\eta \epsilon \bar{R}_\delta$, $T(\eta)$ is the position of the last non-δ element in η; in other words:

$$\text{for any } \eta \epsilon \bar{R}_\delta, T(\eta)=i \leftrightarrow \eta(i)\neq\delta \text{ and } \eta(j)=\delta \text{ for } j>i.$$

In this paper, we will denote bounded data sequences by small greek letters and simply refer to them as sequences. This will not cause any confusion because we will never consider anything but bounded data sequences.

In addition to the operators extended from R_δ to $\bar{R}_\delta$, we may also define operators directly on $\bar{R}_\delta$. In general, an n-ary sequence operator Γ is a transformation $\Gamma:[\bar{R}_\delta]^n \rightarrow \bar{R}_\delta$ where $[\bar{R}_\delta]^n = \bar{R}_\delta \times \bar{R}_\delta \times \ldots \bar{R}_\delta$ is the cartesian product space of n copies of $\bar{R}_\delta$. Two basic unary operators that will be frequently used in this paper are the shift operator Ω^k and the spread operator Θ^r defined by:

$$\Omega^k \xi = \eta \quad \text{and} \quad \Theta^r \xi = \zeta,$$

where

$$\eta(i) = \xi(i-k) \qquad i \in N.$$

$$\zeta(i) = \begin{cases} \xi\left(\frac{i+r}{r+1}\right) & i=1, r+2, 2r+3, \ldots, (n-1)r+n, \ldots \\ \delta & \text{otherwise.} \end{cases}$$

More descriptively, Ω^k inserts k δ-elements at the beginning of a sequence, while Θ^r inserts r δ-elements between every two elements of a sequence. For example if $\xi = a_1, a_2, a_3, a_4, \delta, \delta, \ldots$ then $T(\xi)=4$ and

$$\xi(i) = a_i \qquad 1 \leq i \leq T(\xi)$$

$$\Omega^3\xi = \delta, \delta, \delta, a_1, a_2, a_3, a_4, \delta, \delta, \delta, \ldots$$

$$\Theta^2\xi = a_1, \delta, \delta, a_2, \delta, \delta, a_3, \delta, \delta, a_4, \delta, \delta, \ldots$$

It is easy to verify that the termination function generally satisfies

$$T(\Omega^k\xi) = T(\xi)+k$$

$$T(\Theta^r\xi) = (r+1)T(\xi)-r$$

It is also clear that we can define a sequence operator by combining previously defined sequence operators. For example we might define an operator $\Gamma: \bar{R}_\delta \times \bar{R}_\delta \times \bar{R}_\delta \to \bar{R}_\delta$ as follows:

$$\Gamma(\xi, \eta, \zeta) = \Omega[\xi + \eta * \zeta]$$

where square brackets are used for grouping and parenthesis

for enclosing the arguments of the operator.

We next define a causal operator to be any n-ary sequence operator $\Gamma:[\bar{R}_0]^n \to \bar{R}_0$ which satisfies the causality property in the sense that the i^{th} element of any of its operands can only affect the j^{th} element of its image for $j>i$. In order to formulate this more precisely, assume that for any given sequences $\eta_r \epsilon \bar{R}_0$ $r=1,2,\ldots,n$, the image under Γ is $\xi=\Gamma(\eta_1,..\eta_r,..\eta_n)$. Then Γ is a causal operator if by replacing any operands η_r by another sequence η_r' satisfying

$$\eta_r'(t) = \eta_r(t) \qquad 1 \leq t < i$$

the resulting image $\xi' = \Gamma(\eta_1,..\eta_r',..\eta_n)$ satisfies

$$\xi'(t) = \xi(t) \qquad 1 \leq t \leq i$$

In other words, the value of $\xi(i)$ depends only on the first i-1 elements of η_r, $1 \leq r \leq n$.

Similarly, we may define weakly-causal operators for which the i^{th} element of the image sequence $\xi(i)$ depends only on the the first i elements of the operands η_r, $1 \leq r \leq n$ instead of the first i-1 elements. With this, it is easily seen that the combination $\Gamma^1 \Gamma^2$ (or $\Gamma^2 \Gamma^1$) of a causal operator Γ^1 and a weakly-causal operator Γ^2 is a causal operator. For instance, the shift operator Ω^k is causal

and the spread operator θ^r is weakly causal; hence, the combined operator $\Omega^k \theta^r$ is causal.

2.2. The abstract model.

In order to define the mathematical model used in our verification technique, we define as usual a loop-less multigraph $G(V,E,\varphi_-,\varphi_+)$ to be composed of

(a) a set V of nodes;

(b) a set E of directed edges;

(c) two functions $\varphi_-,\varphi_+:E\to V$. satisfying the condition that for any edge $e\epsilon E$,

$$\varphi_-(e) \neq \varphi_+(e) \qquad (2.2)$$

For each edge $e\epsilon E$, the nodes $\varphi_-(e)$ and $\varphi_+(e)$ are the source and destination node, respectively, of that edge. Clearly, the condition (2.2) prevents any direct loops in the graph. This definition of a multigraph allows any two nodes to be connected by more than one edge in the same direction, a property that may be useful when we represent systolic networks by this abstract model.

As usual in graph terminology, for any node $v\epsilon V$, the edges $\{e;\varphi_-(e)=v\}$ directed out of v are termed the OUT edges of v, while the edges $\{e;\varphi_+(e)=v\}$ directed into v are termed the IN edges of v. Accordingly, the IN-degree and

OUT-degree of v are the number of IN edges and OUT edges of v, respectively. Any node $v \in V$ with IN-degree zero or OUT-degree zero is called a source or a sink, respectively. All other nodes are called interior nodes of G. We shall use the notation V_S, V_T and V_I for the subsets of V containing the source, sink and interior nodes of V, respectively. Of course, the condition $V_S \cup V_T \cup V_I = V$ is always satisfied.

With this notion of a multigraph, we define our abstract systolic model to be composed of the following components.

[A1] A multigraph $G(V,E,\varphi_-,\varphi_+)$.

[A2] A coloring function $col:E \to C_E$, which maps E into a given finite set of colors C_E, and hence assigns a color to each edge in E. The coloring function is assumed to satisfy the condition that the different IN edges of a node have different colors, and correspondingly that the different OUT edges of a node have different colors. Edge colors $y=col(e)$, will be denoted by lower case letters.

[A3] For each edge $e \in E$, a sequence $\xi_e \in \bar{R}_0$ is specified.

[A4] For each interior node $v \in V$ with IN degree m and OUT degree n, we are given n causal m-ary operators $\Gamma_v^i: [\bar{R}_0]^m \to \bar{R}_0$ which specify the "node I/O description". More

specifically, if η^j, j=1,2,...,m and ξ^i, i=1,2,...,n are the sequences associated with the IN and OUT edges of v, respectively, then the n relations

$$\xi^i = \Gamma_v^i(\eta^1,\eta^2,\ldots,\eta^m) \qquad i=1,2,\ldots,n$$

are the I/O description of v. The different IN and OUT edges of v are distinguished in the I/O description by their colors.

Since by condition [A2] all edges terminating at a given node v have different colors, it follows that any edge $e\in E$ is uniquely identified by a pair (y,v), where y=col(e) and $v=\varphi_+(e)$. To simplify the notation, the pair (y,v) will often be written in the form y_v, and the sequence associated with that edge will be identified by the symbol η_v, where we replaced the letter y by its corresponding greek letter η.

For practical applications, it is generally desirable to identify the nodes of the network by appropriate labels which correspond to the problem at hand. This means that we introduce a set L of labels together with a one-to-one function $\psi:V\to L$ from V onto L. In our examples, we usually identify directly the nodes with their labels.

After defining the general abstract model, we next show how it can be used to define a general systolic network.

2.3. The general systolic network.

By giving a physical interpretation to each component in the general abstract model we obtain a general systolic network. The basic idea of this interpretation may be summarized as follows:

Each interior node represents a computational cell and each source/sink node corresponds to an input/output cell for the overall network. To distinguish in our figures the computational cells from the I/O cells, we depict computational cells by circular nodes and I/O cells by square nodes.

Each edge $x_v \epsilon E$ represents a unidirectional communication link between the two cells it connects. The sequence associated with x_v then comprises the data items that appeared on it in consecutive time units. More specifically, if ξ_v is the sequence associated with x_v, then the i^{th} element of ξ_v, namely $\xi_v(i)$ is the data item that appeared on x_v at time t=i units, where t=1 is the time at which the network started its operation.

For an interior node, the node I/O description describes the computation performed by the cell corresponding to that node. We illustrate this with two simple examples:

EX 1: The node shown in figure 4 represents a simple

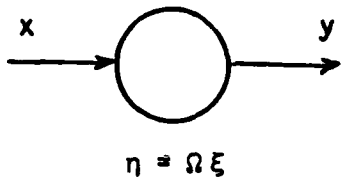


Figure (4)

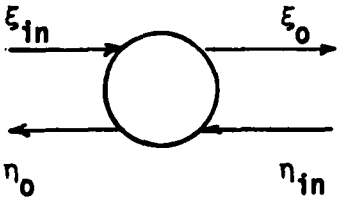


Figure (5)

latch cell which produces at any time t>1 on its output link the same data item that appeared on its input link at time t-1. At time t=1, we have $\eta(1)=\delta$, which corresponds to the fact that at the beginning of the network operation, no specific data item appeared on the output link.

EX 2: The operation of the multiply-add cell mentioned in section 1 and shown in figure 1 may be represented by the following node I/O descriptions:

$$\xi_o = \Omega\ \xi_{in} \qquad (2.3.a)$$

$$\eta_o = \Omega\ (\eta_{in} + w\ .\ \xi_{in}) \qquad (2.3.b)$$

where $w \in R$ is a given real number and ξ_{in}, η_{in}, ξ_o and η_o are the input and output sequences of the node as shown in figure 5.

Since in any practical dynamical system any data item produced by a computational cell at time t depends only on the data provided to that cell at times less than t, we immediately see the importance of the condition imposed in

section 2.2 on the node I/O descriptions, namely that only causal operators in the sense of section 2.1 are used. We also note that with the model described above, the computational power of each cell is not limited to simple arithmetical operations. In other words, a cell could be an intelligent cell that can perform elaborate calculations provided that we can express these calculations in terms of causal operators.

We call "network output sequences" those sequences associated with the IN edges of sink nodes, and "network input sequences" those associated with the OUT edges of source nodes. Then the system of all node I/O descriptions provides a specification of the computation performed by the network in the form of an implicit relation between the network input and output sequences. This relation will be called the "network I/O description".

As a simple example, consider the hypothetical network with the graph shown in figure 6. In this graph, we assume that the edges directed to the left are given the color y and those directed to the right the color x. We also follow the naming convention mentioned in section 2.2 in identifying the different edges in the graph. To complete the network description, a node I/O description has to be specified for each node in the graph. Assume that these are given by the following causal relations:

For node 1: $\xi_2 = \Omega [\xi_1 + \eta_1]$ (2.4.a)

$$\eta_0 = \Omega [\xi_1 * \eta_1] \qquad (2.4.b)$$

For node 2: $\xi_3 = \Omega \xi_2$ (2.5)

For node 3: $\eta_1 = \Omega [\xi_3 * \eta_3]$ (2.6)

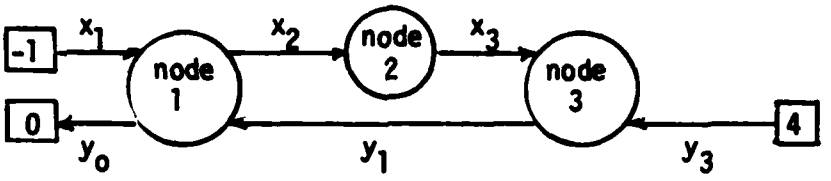


Figure (6)

For this network, η_3 and ξ_1 are the network input sequences and η_0 is the network output sequence. In order to obtain the network I/O description explicitly, we have to solve the equations (2.4), (2.5) and (2.6), that is, we have to obtain an explicit expression for η_0 in terms of ξ_1 and η_3.

Generally, it is very difficult, and sometimes impossible, to derive an explicit solution of the system of node I/O equations. In the next section, we show that this task may be greatly simplified in the case of certain networks with a homogeneous structure.

3. Homogeneous Systolic Networks.

By condition [A2], any edge $e \in E$ is uniquely identified by its color and one of its incident nodes. In fact, we used this already as a convenient means for identifying edges by their color and terminal node. Let $M \subset C_E \times V_I$ be the set of all pairs (y,v), $y \in C_E$, $v \in V_I$, for which there is an edge $e \in E$ with $y=col(e)$ and $v=\varphi_-(e)$. Then the terminal node $u=\varphi_+(e)$ is uniquely given and hence the successor function $\mu: M \to V_I \cup V_T$ is well defined by the association

$$(y,v) \in M,\ y=col(e),\ v=\varphi_-(e) \to \mu(y,v)=\varphi_+(e).$$

In other words, if there exists an edge e with color y and starting node v, then $\mu(y,v)$ is the terminal node of e.

Given a systolic network based on the graph $G=\{V,E,\varphi_-,\varphi_+\}$, a subset $V_I' \subseteq V_I$ of interior nodes is said to be a homogeneous set if:

[H1] All the nodes in V_I' have identical IN and OUT degrees, say m and n, respectively.

[H2] The m colors of the IN edges of any interior node $v \in V_I'$ are identical. So are the n colors of the OUT edges of v. Denote the colors of the IN and OUT edges of v by $y^1, y^2, \ldots, y^m$ and $z^1, z^2, \ldots, z^n$, respectively.

[H3] The node I/O descriptions of any interior node

$v \epsilon V_I'$ are generic in the sense that they may be written in the form:

$$\zeta^i_{\mu(z^i,v)} = \Gamma^i(\eta_v^1, \eta_v^2, \ldots, \eta_v^m) \qquad i=1,2,\ldots,n$$

where Γ^i, $i=1,2,\ldots,n$ are given n-ary operators which are independent of the particular node in V_I', μ is the successor function defined earlier in this section and η_v^j $j=1,2,\ldots,m$ and $\zeta^i_{\mu(z^i,v)}$ $i=1,2,..,n$ are the sequences associated with the IN and OUT edges of v, respectively.

A network is said to be homogeneous if the set of interior nodes V_I in its graph G is a homogeneous set. More generally, if there exists a partition $V_I = V_I^1 \cup V_I^2 \cup \ldots \cup V_I^k$ of V_I into k non-empty homogeneous subsets $V_I^1, V_I^2, \ldots, V_I^k$, then the network is said to be k-partially homogeneous.

The main advantage of having a homogeneous (or partially homogeneous) network is that the resulting system of equations has a repetitive pattern, which, in many cases, allows us to obtain an analytical solution to the system. This should become clearer as we proceed with the different examples.

To verify the operation of a systolic network, we are generally interested in its behavior for specific inputs, that is we wish to find the form of the network output sequences for specific network input sequences. This is usually accomplished by substituting the given input sequences in the network I/O description and manipulating the resulting equations to obtain the description of the network output sequences.

As a first example of our verification technique, we consider again the 1-D convolution network described in section 1. The graph of this network is shown in figure 7, where we assumed that the edges directed to the left have the color 's', while those directed right have the color 'p'. The nodes of the graph are identified by the integers $-1,0,1,2,\ldots,k+1,k+2$, where nodes -1 and $k+2$ are source nodes, nodes 0 and $k+1$ sink nodes, and nodes 1 through k interior nodes. The successor function is defined for any interior node $i=1,2,\ldots,k$ by

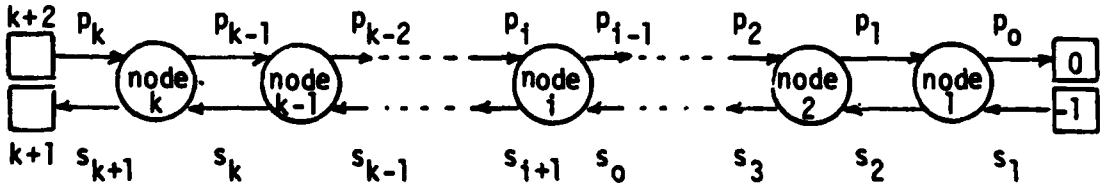


Figure (7)

$$\mu(y,i) = \begin{cases} i+1 & \text{if } y=s \\ i-1 & \text{if } y=p \end{cases}$$

Our goal is to verify that the network indeed produces the results of equation (1.1) for the network input sequences described by

$$\sigma_1 = \Omega^{k-1}\,\theta_\iota \tag{3.1.a}$$

$$\pi_k = \theta_\xi \tag{3.1.b}$$

where

$$T(\iota)=n-(k-1) \quad ,\iota(t)=0$$

$$T(\xi)=n \quad ,\xi(t)=x_t$$

The I/O description of a typical interior node i in the graph, $1 \le i \le k$, is given by the following causal relations

$$\pi_{i-1} = \Omega\,\pi_i \tag{3.2.a}$$

$$\sigma_{i+1} = \Omega\,[\sigma_i + w_i \cdot \pi_i\,] \tag{3.2.b}$$

This system of difference equations is easily solved. First, note that the solution of (3.2.a) obviously is

$$\pi_i = \Omega^{k-i}\,\pi_k \tag{3.3}$$

By substituting this in (3.2.b) we obtain

$$\sigma_{i+1} = \Omega\,\sigma_i + w_i \cdot [\,\Omega^{k-i+1}\,\pi_k\,] \tag{3.4}$$

The solution of (3.4) is then given by lemma 1 in the appendix as:

$$\sigma_{k+1} = \Omega^k \sigma_1 + \sum_{j=1}^{k} \Omega^{j-1} [w_{k-j+1} \cdot \Omega^{k-(k-j+1)+1} \pi_k]$$

$$= \Omega^k \sigma_1 + \sum_{j=1}^{k} \Omega^{2j-1} [w_{k-j+1} \cdot \pi_k] \qquad (3.5)$$

This is the I/O description for the network.

In order to find the specific form of the output sequence σ_{k+1} for the input sequences (3.1), we substitute these sequences into (3.5) and obtain

$$\sigma_{k+1} = \Omega^{2k-1} \Theta \iota + \sum_{j=1}^{k} \Omega^{2j-1} [w_{k-j+1} \cdot \Theta \xi]$$

By the properties P1, P2, P3 and P4 in the appendix, this may be rewritten as

$$\sigma_{k+1} = \Omega^{2k-1} \Theta \iota + \Omega \sum_{j=1}^{k} \Omega^{2(j-1)} \Theta [w_{k-j+1} \cdot \xi]$$

$$= \Omega^{2k-1} \Theta \iota + \Omega \Theta \sum_{j=1}^{k} \Omega^{j-1} [w_{k-j+1} \cdot \xi]$$

$$= \Omega^{2k-1} \Theta \iota + \Omega \Theta \sum_{j=1}^{k} \Omega^{j-1} \eta_j$$

where $T(\eta_j)=T(\xi)=n$ and $\eta_j(t)=w_{k-j+1}\ \xi(t)=w_{k-j+1}\ x_t$.

Finally, applying P5 of the appendix we find:

$$\sigma_{k+1} = \Omega^{2k-1} \Theta \iota + \Omega \Theta \Omega^{k-1} \eta$$

$$= \Omega^{2k-1} \Theta [\iota + \eta]$$

$$= \Omega^{2k-1} \Theta \eta \qquad (3.6)$$

where η is defined by:

$$T(\eta) = n-(k-1)$$

$$\eta(t) = \sum_{j=1}^{k} \eta_j(t+k-j) \qquad 1 \leq t \leq T(\eta)$$

$$= \sum_{j=1}^{k} w_{k-j+1}\, x_{t+k-j} \qquad 1 \leq t \leq T(\eta)$$

$$= \sum_{q=1}^{k} w_q\, x_{t+q-1} \qquad 1 \leq t \leq T(\eta)$$

In the last line, the summation index was changed to q=k-j+1 in order to provide for the same expression as in (1.1).

Evidently, equation (3.6) represents the output of the array in a clear and precise form; it indicates that after an initial period of 2k-1 time units, the elements $\eta(t)=y_t$, $1 \leq t \leq n-(k-1)$, will appear on the output link, each separated from the other by one time unit.

A variation of the above 1-D convolution network may be obtained by defining the I/O description of each node in the network to be given by (3.2.a) and (3.2.b) with the + operation replaced by the $\oplus$ operation defined by (2.1). By a similar analysis, it can be shown that the output of the modified network is described by

$$\sigma_{k+1} = \Pi \; \Theta \; \eta'$$

where $T(\eta') = n+k-1$ and

$$\eta'(t) = \begin{cases} \sum_{j=1}^{t} w_{k-j+1}\, x_{t-j+1} & 1 \le t \le k-1 \\ \sum_{j=1}^{k} w_{k-j+1}\, x_{t-j+1} & k \le t \le n \\ \sum_{j=t-n+1}^{k} w_{k-j+1}\, x_{t-j+1} & n+1 \le t \le T(\eta'). \end{cases}$$

In the previous example we applied our technique to a homogeneous network. The technique is equally applicable to k-partially homogeneous networks if k is reasonably small. In that case, a system of difference equations is formed by writing the generic I/O description for a typical node from each homogeneous subset of interior nodes V_I^i, i=1,2,...,k. The network I/O description is then obtained by solving this system of equations. The back substitution network and the sorting networks discussed in sections 5 and 6 are examples of 2-partially homogeneous networks. The LU decomposition network described in [1] is a 4-partially homogeneous network that can be verified by the same technique.

Finally, we note that the explicit derivation of the network I/O description depends on our ability to solve the resulting system of difference equations. However, even if these equations cannot be solved explicitly, we may still verify the operation of the network if we have an idea about the network behavior and consequently about the sequences on the different edges of the graph. In fact, we need to show

only that for the given input sequences, the expected sequences satisfy the system of difference equations. We demonstrate this procedure in section 6 by verifying the operation of a sorting network for which we could not solve the system of equations explicitly.

4. A band matrix multiplication network.

In [1], Kung and Leiserson suggested a systolic network for the computation of the product of two band matrices C=A*B, where both A and B have lower bandwidth k_1 and upper bandwidth k_2. In this section, we shall consider only the case $k_1=k_2=k$ and prove formally that the suggested network indeed produces the product matrix C. Moreover, the sequence notation used in the verification procedure will provide an accurate representation of the I/O data including the input timing required for proper operation and the timing of the output data.

In figure 8.a we show the directed graph of the matrix multiplication network. The nodes of the graph are regularly laid out so that each node can be labeled by a pair (i,j) of integers, where i and j are the relative position of the node with respect to the two perpendicular axes shown in the figure. The set of colors C_E has three elements, namely p, r and s, and the coloring function col() maps the edges directed to the south-west, south-east and north to the colors p,r and s, respectively.

The network is homogeneous; it consists of only one type of computational cell, namely the multiply-add type cell shown in figure 8.b. Its generic I/O description is given by the causal relations:

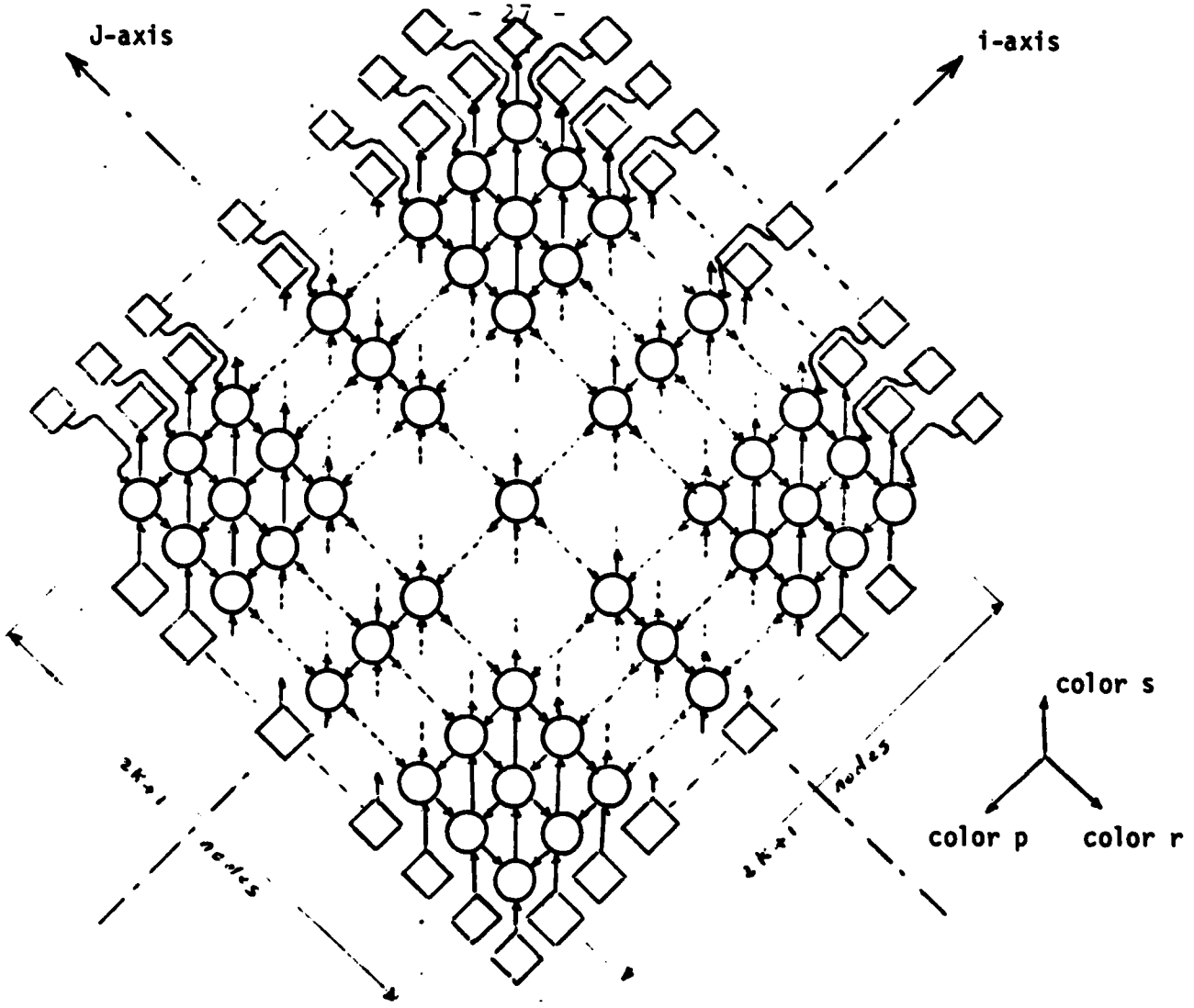


Figure (8.a)

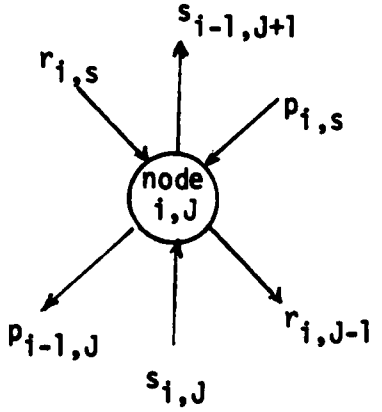


Figure (8.b)

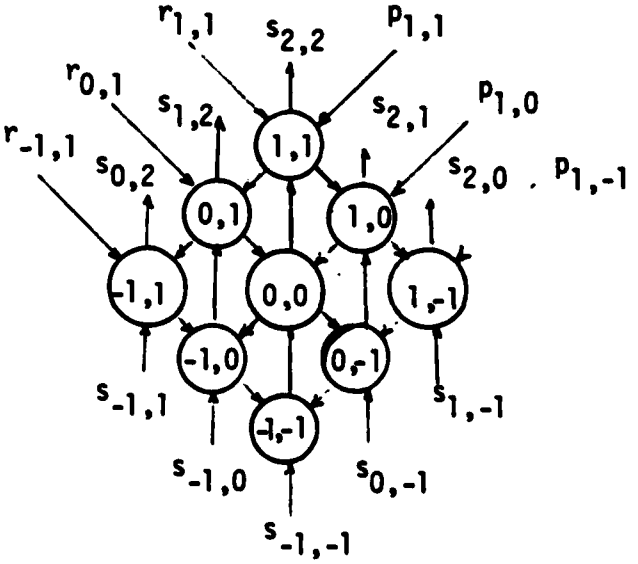


Figure (8.c)

$$\rho_{i,j-1} = \Omega\ \rho_{i,j} \qquad (4.1.a)$$

$$\pi_{i-1,j} = \Omega\ \pi_{i,j} \qquad (4.1.b)$$

$$\sigma_{i+1,j+1} = \Omega[\sigma_{i,j} + \rho_{i,j} * \pi_{i,j}] \qquad (4.1.c)$$

In line with the definition of homogeneous networks, this description is valid for any cell (i,j), $-k \leq i,j \leq k$.

As an illustration of the network topology and its different data streams, we show in figure 8.c the general network for the special case k=1, that is for the case of two tridiagonal matrices A and B. In the figure, the source/sink cells were omited for clarity.

In order to obtain the I/O description of the network, we have to solve the system of difference equations (4.1), and express the network output sequences $\sigma_{q,k+1}$ and $\sigma_{k+1,q}$, $-(k-1) \leq q \leq k+1$ in terms of the network input sequences $\rho_{u,k}$, $\pi_{k,u}$, $\sigma_{-k,u}$ and $\sigma_{u,-k}$, $-k \leq u \leq k$. For this, consider first the simple equations (4.1.a) and (4.1.b) which have the solutions

$$\rho_{i,j} = \Omega^{k-j}\ \rho_{i,k}$$

$$\pi_{i,j} = \Omega^{k-i}\ \pi_{k,j}$$

By substituting these values into (4.1.c) we obtain

$$\sigma_{i+1,j+1} = \Omega\ [\sigma_{i,j} + \Delta_{i,j}] \qquad (4.2)$$

where $\Delta_{i,j} = \Omega^{k-j}\ \rho_{i,k} * \Omega^{k-i}\ \pi_{k,j}$

By an inductive argument similar to the one given in the appendix for lemma 1, it is easily shown that for $-(k-1) \leq i,j \leq k+1$, (4.2) has the solution:

$$\sigma_{i,j} = \begin{cases} \Omega^{i+k}\, \sigma_{-k,j-i-k} + \sum_{q=1}^{k+i} \Omega^{q}\, \Delta_{i-q,j-q} & i \leq j \\ \Omega^{j+k}\, \sigma_{i-j-k,-k} + \sum_{q=1}^{k+j} \Omega^{q}\, \Delta_{i-q,j-q} & i > j \end{cases}$$

With the definition of $\Delta_{i,j}$ and properties P1 and P4 we find the network output sequences to be

$$\sigma_{i,k+1} = \Omega^{i+k}\, \sigma_{-k,1-i} + \sum_{q=1}^{i+k} [\, \Omega^{2q-1}\, \rho_{i-q,k} * \Omega^{2q+k-i}\, \pi_{k,k-q+1}] \qquad -(k-1) \leq i \leq k+1 \qquad (4.3.a)$$

$$\sigma_{k+1,j} = \Omega^{k+j}\, \sigma_{1-j,-k} + \sum_{q=1}^{k+j} [\, \Omega^{2q+k-j}\, \rho_{k-q+1,k} * \Omega^{2q-1}\, \pi_{k,j-q}] \qquad -(k-1) \leq j \leq k \qquad (4.3.b)$$

These are the network I/O descriptions. Of course, the network is not expected to produce the elements of the product matrix C unless the elements of the matrices $A=\{a_{i,j}\}$ and $B=\{b_{i,j}\}$ are fed into the proper input links of the network with the right timing. We will now prove that the network output sequences will contain the elements of C if the input sequences are specified as follows:

$$\rho_{u,k} = \Omega^{2(k+u)}\ \theta^2\ \alpha_u \qquad -k \le u \le k \qquad (4.4.a)$$

$$\pi_{k,u} = \Omega^{2(k+u)}\ \theta^2\ \beta_u \qquad -k \le u \le k \qquad (4.4.b)$$

$$\sigma_{-k,u} = \Omega^{2(2k+u)}\ \theta^2\ \iota_u \qquad -k \le u \le k \qquad (4.4.c)$$

$$\sigma_{u,-k} = \Omega^{2(2k+u)}\ \theta^2\ \iota_u \qquad -k < u \le k \qquad (4.4.d)$$

where

$$T(\beta_u)=T(\alpha_u)=n, \quad T(\iota_u)=n-(k+u), \quad \iota_u(t)=0$$

and the sequences β_u, α_u are defined as follows:

For $u<0$

$$\alpha_u(t) = \begin{cases} 0 & 1 \le t \le -u \\ a_{t,t+u} & -u < t \le n \end{cases} \qquad (4.5.a)$$

$$\beta_u(t) = \begin{cases} 0 & 1 \le t \le -u \\ b_{t+u,t} & -u < t \le n \end{cases} \qquad (4.5.b)$$

For $u \ge 0$

$$\alpha_u(t) = \begin{cases} a_{t,t+u} & 1 \le t \le n-u \\ 0 & n-u < t \le n \end{cases} \qquad (4.5.c)$$

$$\beta_u(t) = \begin{cases} b_{t+u,t} & 1 \le t \le n-u \\ 0 & n-u < t \le n \end{cases} \qquad (4.5.d)$$

Roughly speaking, the input link $p_{k,u}$, $-k \le u \le k$, contains

the u^{th} off-diagonal of the matrix B, while the input link $r_{u,k}$, $-k \leq u \leq k$, contains the $(-u)^{th}$ off diagonal of the matrix A. Of course the exact timing of the input data is defined by the formulas (4.4).

For the sake of breviety, we cosider here only the equations (4.3.a) and show that the output links $s_{i,k+1}$, $-(k-1) \leq i \leq k+1$ will carry the elements in the lower band of the product matrix C=A*B, including the diagonal. By a similar procedure, one can use (4.3.b) to show that the links $s_{k+1,j}$, $-(k-1) \leq j \leq k$ will carry the upper band of C.

By introducing the specifications (4.4) of the network input sequences into (4.3.a), we obtain for $-(k-1) \leq i \leq k+1$ the following formula:

$$\sigma_{i,k+1} = \bar{\iota}_i + \sum_{q=1}^{k+i} [\Omega^{2k+2i-1}\, \theta^2\, \alpha_{i-q} * \Omega^{5k-i+2}\, \theta^2\, \beta_{k-q+1}]$$

$$= \bar{\iota}_i + \Omega^{2k+2i-1} \sum_{q=1}^{k+i} \theta^2\, \alpha_{i-q} * \Omega^{3(k-i+1)}\, \theta^2\, \beta_{k-q+1}$$

$$= \bar{\iota}_i + \Omega^{2k+2i-1}\, \theta^2 \sum_{q=1}^{k+i} [\, \alpha_{i-q} * \Omega^{k-i+1}\, \beta_{k-q+1}]$$

where $\bar{\iota}_i = \Omega^{5k-i+2}\, \theta^2\, \iota_{1-i}$. With property P7 the product term becomes

$$\sigma_{i,k+1} = \bar{\iota}_i + \Omega^{2k+2i-1}\, \theta^2 \sum_{q=1}^{k+i} \Omega^{k-i+1}\, \gamma_i^q \qquad (4.6)$$

where $T(\gamma_i^q) = n-(k-i+1)$ and

$$\gamma_i^q(t) = \alpha_{i-q}(t+k-i+1) * \beta_{k-q+1}(t)$$

Simplifying (4.6) and using the definition of $\bar{\iota}_i$, we find that

$$\sigma_{i,k+1} = \Omega^{5k-i+2}\,\theta^2\,\iota_{1-i} + \Omega^{5k-i+2}\,\theta^2 \sum_{q=1}^{k+i} \gamma_i^q$$

$$= \Omega^{5k-i+2}\,\theta^2\,[\iota_{1-i} + \eta_i]$$

where $T(\eta_i) = n-(k-i+1)$ and

$$\eta_i(t) = \sum_{q=1}^{k+i} \gamma_i^q(t) \qquad -(k-1)\leq i\leq k+1$$

$$= \sum_{q=1}^{k+i} \alpha_{i-q}(t+k-i+1) * \beta_{k-q+1}(t) \qquad -(k-1)\leq i\leq k+1$$

(4.7)

Finally, from the definition of ι_{1-i} we obtain that

$$\sigma_{i,k+1} = \Omega^{5k-i+2}\,\theta^2\,\eta_i \qquad -(k-1)\leq i\leq k+1 \qquad (4.8)$$

Equation (4.8) describes the timing of the output data on any link $s_{i,k+1}$, $-(k-1)\leq i\leq k+1$. It indicates that on $s_{i,k+1}$, there will be an initial set up time of $5k-i+2$ units, after which the elements $\eta_i(t)$, $t=1,2,..,n-(k-i+1)$ will appear separated each from the other by two time units. We still need to show that $\eta_i(t)=c_{t+k-i+1,t}$, that is that $s_{i,k+1}$ carries the $(k-i+1)^{st}$ sub diagonal of the matrix C.

To evaluate $\eta_i(t)$ from (4.7), we use the definitions (4.5) to write $\alpha_{i-q}(t+k-i+1)$ and $\beta_{k-q+1}(t)$ for the values of t between 1 and n-(k-i+1), which are the values of t assumed in (4.7). The resulting formulas are:

$$\alpha_u(t+d) = \begin{cases} 0 & \text{if } u<0 \text{ and } 1\le t\le q-(k+1) \\ a(t,i,q) & \text{if } u<0 \text{ and } q-(k+1)<t\le n-d \\ a(t,i,q) & \text{if } u\ge 0 \text{ and } 1\le t\le n+q-(k+1) \\ 0 & \text{if } u\ge 0 \text{ and } n+q-(k+1)<t\le n-d \end{cases} \qquad (4.9.a)$$

$$\beta_v(t) = \begin{cases} 0 & \text{if } v<0 \text{ and } 1\le t\le q-(k+1) \\ b(t,q) & \text{if } v<0 \text{ and } q-(k+1)<t\le n-d \\ b(t,q) & \text{if } v\ge 0 \text{ and } 1\le t\le n+q-(k+1) \\ 0 & \text{if } v\ge 0 \text{ and } n+q-(k+1)<t\le n-d \end{cases} \qquad (4.9.b)$$

where, for simplicity, we introduced the notation

$u = i-q$, $v = k-q+1$, $d = (k+1)-i$,

$a(t,i,q) = a_{t+d,t+d+u}$ and $b(t,q) = b_{t+v,t}$.

which will be used repeatedly in the remainder of this section.

It is clear from (4.9) that the evaluation of $\eta_i(t)$ by (4.7) is non-trivial and depends on the relative values of i and q. For this purpose, we consider two different cases:

Case 1: If $-(k-1) \leq i \leq 0$.

In this case and for $1 \leq q \leq k+i$, the inequalities $u=i-q<0$ and $v=k-q+1>0$ always hold. Moreover, we have $q-(k+1) \leq 0$ and $n+q-(k+1)>n-d$. Accordingly, we can use the above conditions to determine the appropriate values of $\alpha_u(t+d)$ and $\beta_v(t)$ from (4.9), and with these in (4.7) we obtain the formula:

$$\eta_i(t) = \sum_{q=1}^{k+i} a_{t+d,t+k+1-q} \; b_{t+k+1-q,t} \qquad 1 \leq t \leq n-d$$

By changing the summation index to $j=t+k+1-q$ this is indeed

$$\eta_i(t) = \sum_{j=t+d-k}^{t+k} a_{t+d,j} \; b_{j,t} \qquad 1 \leq t \leq n-d \qquad (4.10)$$

Case 2: If $1 \leq i \leq k+1$.

In this case we always have $u=i-q \leq v=k-q+1$. Accordingly, we divide the sum in (4.7) into the three partial sums

$$\sum_{q=1}^{k+i} = \sum_{q=1}^{i} + \sum_{q=i+1}^{k} + \sum_{q=k+1}^{k+i}$$

For simplicity, we refer to these three sums as Σ_1, Σ_2 and Σ_3, respectively, and evaluate them separately.

In the case of $\Sigma_1 = \sum_{q=1}^{i} \gamma_i^q(t)$, we note that the condition $1 \leq q \leq i$ implies that $v \geq u \geq 0$. Hence, by (4.9) we have

$$\gamma_i^q(t) = \begin{cases} a(t,i,q)\ b(t,q) & \text{if } 1 \le t \le n+q-(k+1) \\ 0 & \text{if } n+q-(k+1) < t \le n-d \end{cases}$$

By standard rules of operations with summation symbols, Σ_1 can be expressed as

$$\Sigma_1 = \begin{cases} \sum_{q=1}^{i} a(t,i,q)\ b(t,q) & \text{if } 1 \le t \le n-k \\ \sum_{q=t-n+k+1}^{i} a(t,i,q)\ b(t,q) & \text{if } n-k < t \le n-d \end{cases} \qquad (4.11)$$

We turn next to $\Sigma_2 = \sum_{q=i+1}^{k} \gamma_i^q(t)$. In this case, we have $u<0\le v$, $q-(k+1)<0$ and $n+q-(k+1)>n-d$. Hence, from (4.9) it follows that

$$\gamma_i^q(t) = a(t,i,q)\ b(t,q) \qquad 1 \le t \le n-d$$

which gives directly

$$\Sigma_2 = \sum_{q=i+1}^{k} a(t,i,q)\ b(t,q) \qquad 1 \le t \le n-d \qquad (4.12)$$

Finally, in the case of Σ_3 the inequality $u\le v<0$ holds. Therefore, we have

$$\gamma_i^q(t) = \begin{cases} 0 & \text{if } 1 \le t \le q-(k+1) \\ a(t,i,q)\ b(t,q) & \text{if } q-(k+1) < t \le n-d \end{cases}$$

which gives

$$\Sigma_3 = \begin{cases} \sum_{q=k+1}^{k+t} a(t,i,q)\ b(t,q) & \text{if } 1 \le t \le i \\ \sum_{q=k+1}^{k+i} a(t,i,q)\ b(t,q) & \text{if } i < t \le n-d \end{cases} \qquad (4.13)$$

Now $\eta_i(t)$ is obtained by adding the sums (4.11), (4.12) and (4.13) on three different intervals for t. This sum is given by

$$\eta_i(t) = \begin{cases} \sum_{q=1}^{k+t} a(t,i,q)\ b(t,q) & 1 \le t \le i \\ \sum_{q=1}^{k+i} a(t,i,q)\ b(t,q) & i < t \le n-k \\ \sum_{q=t-n+k+1}^{k+i} a(t,i,q)\ b(t,q) & n-k < t \le n-d \end{cases}$$

By changing the summation index to j=t+k+1-q and substituting the appropriate values for a(t,i,q) and b(t,q) we obtain

$$\eta_i(t) = \begin{cases} \sum_{j=1}^{t+k} a_{t+d,j}\, b_{j,t} & 1 \le t \le i \\ \sum_{j=t+d-k}^{t+k} a_{t+d,j}\, b_{j,t} & i < t \le n-k \\ \sum_{j=t+d-k}^{n} a_{t+d,j}\, b_{j,t} & n-k < t \le n-d \end{cases}$$

Note that the above formula for $\eta_i(t)$ is valid for $1 \le i \le k+1$ while (4.10) is valid for $-(k-1) \le i \le 0$. These two formulas are equivalent to those resulting from multiplying the two band matrices A and B, which proves that for $t=1,2,\ldots,n-(k-i+1)$ and $-(k-1) \le i \le k+1$, we have indeed

$$\eta_i(t) = c_{t+d,t} = c_{t+k-i+1,t}.$$

5. A back substitution network.

In this section, we apply our verification technique to a systolic network that contains two different types of computational cells, namely the back-substitution network suggested in [8]. This network performs the back substitution operation to solve the linear system of equations

$$L\,u = y \tag{5.1}$$

where L is an nxn non-singular, banded, lower triangular matrix with the band width k+1, and y is a given n-dimensional vector. The solution of the system (5.1) is given by the formula:

$$u_i = \begin{cases} y_i \;/\; l_{i,i} & i=1 \\ (y_i - \sum_{j=1}^{i-1} l_{i,i-j}\, u_{i-j}) \;/\; l_{i,i} & 2 \le i \le k \\ (y_i - \sum_{j=1}^{k} l_{i,i-j}\, u_{i-j}) \;/\; l_{i,i} & k < i \le n \end{cases}$$

where $l_{i,j}$ is the $(i,j)^{th}$ element of the matrix L, and y_i and u_i are the i^{th} elements of the vectors y and u, respectively.

Figure 9 shows the graph of the suggested network. It is a 2-partially homogeneous network, composed of k multiply/add (M/A) type cells, and one subtract/divide (S/D) cell. The computational cells are labeled by integers such that the cells 1 through k are of the M/A type, and the cell

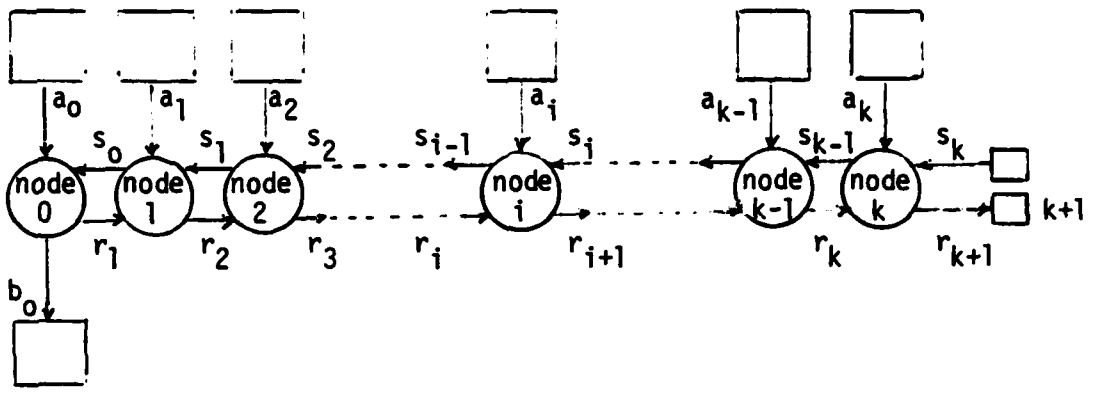


Figure (9)

0 is the S/D cell. As for the I/O cells, we must be careful to assign labels to the sink cells because these labels will be used to identify the network output links. The labels given to source nodes are immaterial as they do not affect the verification procedure, and consequently are not shown in figure 9.

In the regular layout shown in figure 9, the edges directed to the south, north, east and west are given the colors a,b,r and s, respectively. The set V_I of interior nodes in G is divided into two homogeneous subsets $V_I^1=\{0\}$ and $V_I^2=\{i:i=1,2,\ldots,k\}$. The operation of the cell represented by node '0' is described by the causal relation

$$\rho_1 = \Omega\,[[\beta_0 - \sigma_0] + \alpha_0] \qquad (5.2)$$

and the operation of any M/A cell represented by a node i, $1 \leq i \leq k$, is described by the generic I/O description

$$\rho_{i+1} = \Omega\,\rho_i \qquad i=1,2,\ldots,k \qquad (5.3.a)$$

$$\sigma_{i-1} = \Omega[\sigma_i \oplus \alpha_i\,\rho_i] \qquad i=1,2,\ldots,k \qquad (5.3.b)$$

where the $\oplus$ was defined by (2.1).

To solve the system of difference equations (5.2), (5.3.a/b), we first write the solution of (5.3.a) as

$$\rho_i = \Omega^{i-1} \rho_1 \qquad 1<i\leq k+1 \qquad (5.4)$$

from which we find that

$$\rho_{k+1} = \Omega^k \rho_1 \qquad (5.5)$$

Substitution of (5.4) into (5.3.b) then gives

$$\sigma_{i-1} = \Omega [\sigma_i \oplus \Delta_i] \qquad (5.6)$$

where $\Delta_i = \alpha_i * (\Omega^{i-1} \rho_1)$. Using an inductive argument similar to that in the appendix for the proof of lemma 1, we can show that the solution of (5.6) is

$$\sigma_0 = \Omega^k \sigma_k \oplus \sum_{j=1}^{k}{}' \Omega^j [\alpha_j * \Omega^{j-1} \rho_1] \qquad (5.7)$$

where Σ' is defined by $\sum_{j=1}^{k}{}' \eta_j = \eta_1 \oplus \eta_2 \oplus \ldots \oplus \eta_k$.

For given ρ_1, the network output sequence ρ_{k+1} is easily obtained from (5.5). The next step will be to eliminate σ_0 from (5.2) and (5.7) and to obtain ρ_1 explicitly in terms of the network input sequences σ_k, β_0 and α_j, $j=0,1,\ldots,k$. Unfortunately, if we try to solve (5.2) and (5.7) simultaneously, we will obtain a recursive equation in ρ_1, which is very difficult to manipulate in general. For

this reason, we consider only specific forms of the network input sequences, namely those required for the proper operation of the network. They are given by

$$\alpha_i = \Omega^{k-i} \ominus \lambda_i \qquad i=0,1,\ldots,k \tag{5.8.a}$$

$$\beta_0 = \Omega^k \ominus \eta \tag{5.8.b}$$

$$\sigma_k = \ominus \iota \tag{5.8.c}$$

with $T(\lambda_i) = T(\iota) = T(\eta) = n$ and

$$\lambda_i(t) = \begin{cases} 0 & 1 \leq t \leq i \\ 1_{t,t-i} & i < t \leq n \end{cases}$$

$$\eta(t) = y_t \qquad 1 \leq t \leq n$$

$$\iota(t) = 0 \qquad 1 \leq t \leq n$$

Substituting (5.8) into (5.2) and (5.7), we find that

$$\rho_1 = \Omega\,[[\Omega^k \ominus \eta - \sigma_0] + \Omega^k \ominus \lambda_0] \tag{5.9.a}$$

$$\sigma_0 = \Omega^k \ominus \iota \oplus \sum_{j=1}^{k}{}' [\Omega^k \ominus \lambda_j * \Omega^{2j-1}\,\rho_1] \tag{5.9.b}$$

Since $\delta - x = \delta$ for any $x \in R_\delta$, (5.9.a) implies the existence of a sequence ξ such that

$$\rho_1 = \Omega^{k+1} \ominus \xi \tag{5.10}$$

whence, by (5.9.b), we find that

$$\sigma_0 = \Omega^k \ominus \iota \oplus \sum_{j=1}^{k}{}' [\Omega^k \ominus \lambda_j * \Omega^{2j+k} \ominus \xi]$$

$$= \Omega^k \ominus [\iota \oplus \sum_{j=1}^{k} [\lambda_j * \Omega^j \xi]]$$

where we used property P2 to interchange Ω^{2j} and $\ominus$. If in addition we let

$$\gamma = \iota \oplus \sum_{j=1}^{k} [\lambda_j * \Omega^j \xi] \tag{5.11}$$

then we can substitute for σ_0 and ρ_1 in (5.9.a) and obtain

$$\Omega^{k+1} \ominus \xi = \Omega [[\Omega^k \ominus \eta - \Omega^k \ominus \gamma] + \Omega^k \ominus \lambda_0]$$

which reduces to

$$\xi = [\eta - \gamma] + \lambda_0 \tag{5.12}$$

For an explicit description of the sequence γ, we need to examine (5.11) more closely. We start by applying property P7 to the product term in (5.11), namely

$$\lambda_j * \Omega^j \xi = \Omega^j \mu_j$$

where

$$T(\mu_j) = \min\{ T(\lambda_j)-j , T(\xi)) \leq n-j \tag{5.13.a}$$

and

$$\mu_j(t) = \lambda_j(t+j) * \xi(t) \tag{5.13.b}$$

This enables us to rewrite (5.11) as

$$\gamma = \iota \oplus \sum_{j=1}^{k} \Omega^j \mu_j \tag{5.14}$$

From (5.14) and the definition of the '+' operator, we conclude that $T(\gamma) = \max\{T(\iota) , T(\mu_j)+j\} = n$, and consequently from (5.12) that

$$T(\xi) = \min\{T(\eta) , T(\gamma) , T(\lambda_0)\} = n.$$

Using this in (5.13.a) we easily see that $T(\mu_j) = n-j$. Now, we apply property P6 to (5.14) and explicitly describe γ by

$$T(\gamma) = T(\iota) = n$$

and

$$\gamma(t) = \begin{cases} 0 & t=1 \\ \sum_{j=1}^{t-1} \mu_j(t-j) & t=2,3,\ldots,k \\ \sum_{j=1}^{k} \mu_j(t-j) & t=k+1,k+2,\ldots,n \end{cases}$$

Finally, with these specific descriptions of η, λ_0 and γ, we directly find the explicit form of the sequence ξ in (5.12) to be

$$\xi(t) = (\eta(t) - \gamma(t)) \; / \; \lambda_0(t)$$

that is

$$\xi(t) = \begin{cases} y_t \;/\; l_{t,t} & t=1 \\ \left(y_t - \sum_{j=1}^{t-1} \xi(t-j)\; l_{t,t-j}\right) / \; l_{t,t} & 2 \leq t \leq k \\ \left(y_t - \sum_{j=1}^{k} \xi(t-j)\; l_{t,t-j}\right) / \; l_{t,t} & k+1 \leq t \leq n \end{cases}$$

A comparision of this expression with the formula given in the beginning of the section for the solution of (5.1) shows readily that

$$\rho_{k+1} = \Omega^{2k+1} \; \theta \; \xi$$

where $T(\xi) = n$ and $\xi(t) = u_t$.

6. A sorting network

The sorting network [2,9] described here accepts an indexed set $X=\{x_1,\ldots,x_k\}$ of k different real numbers, $x_i \epsilon R$, $i \epsilon K=\{1,\ldots,k\}$, and produces as output the same numbers sorted in ascending order. Figure 10 shows the general graph of the network and the labels given to each node. In the figure, the edges directed to the right and left are colored s and p, respectively.

For any $j \epsilon K$, let $y_1,\ldots,y_j$ be the result of sorting the j elements $x_1,\ldots,x_j$ of X in ascending order. Then for all (i,j) of $D=\{(i,j) \epsilon K \times K;\ 1 \leq i \leq j \leq k\}$, the ranking function $f_x: D \to X$ is defined by $f_x(i,j)=y_i$.

With this, we will prove that if the network input sequence π_k is given by

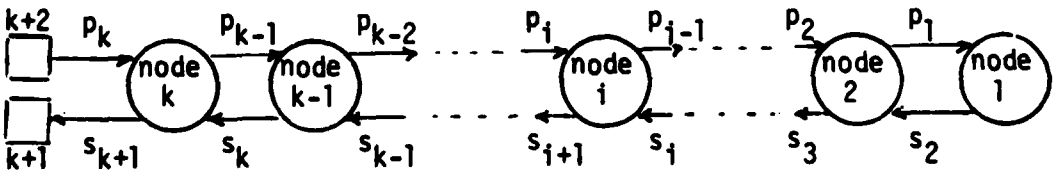


Figure (10)

$$\pi_k = \Theta\ \xi \tag{6.1}$$

where $T(\xi) = k$ and $\xi(t) = x_t$, then the network output sequence σ_{k+1} has the form

$$\sigma_{k+1} = \Omega^{2k-1}\ \Theta\ \eta \tag{6.2}$$

where $T(\eta) = k$ and $\eta(t) = f_x(t,k)$.

The network considered in figure 10 is a 2-partially homogeneous network. The cell labeled '1' is a simple latch cell whose operation is described by

$$\sigma_2 = \Omega\ \pi_1 \tag{6.3.a}$$

while the I/O description of the cells i=2,...,k is given by

$$\pi_{i-1} = \Omega\ \max_\delta(\pi_i, \sigma_i) \tag{6.3.b}$$

$$\sigma_{i+1} = \Omega\ \min_\delta(\pi_i, \sigma_i) \tag{6.3.c}$$

where $\max_\delta$ and $\min_\delta$ were defined in section 2.1. In other words, the cells i=2,...,k are comparision cells which operate as follows: At any time t, if neither one of the two inputs $\sigma_i(t)$ or $\pi_i(t)$ is a don't care element δ, then the cell compares the two inputs, and produces as output at time t+1, the largest and the smallest numbers on the links p_{i-1} and s_{i+1} respectively. However, if any of the inputs is δ, then the cell acts as a simple latch cell, that is, if $\sigma_i(t)=\delta$ or $\pi_i(t)=\delta$ then

$$\pi_{i-1}(t+1) = \pi_i(t) \quad \text{and} \quad \sigma_{i+1}(t+1) = \sigma_i(t)$$

To obtain the network I/O description, the system of equations (6.3.a/b/c) should be solved for σ_{k+1}. However, the recursive nature of (6.3.b) and (6.3.c) makes this very difficult, if not impossible. One possible alternative is to suggest a tentative value for the sequences π_i and σ_i, and then to verify that these suggested solutions indeed satisfy (6.3). Of course, any assumed value for π_i should reduce to the input sequence (6.1) for i=k.

Let us assume that π_i and σ_i are given by

$$\pi_i = \Omega^{k-i} \ominus \alpha_i \qquad 1 \leq i \leq k \qquad (6.4.a)$$

$$\sigma_i = \Omega^{k+i-2} \ominus \beta_i \qquad 2 \leq i \leq k+1 \qquad (6.4.b)$$

where $T(\alpha_i) = T(\beta_i) = k$,

$$\alpha_i(t) = \begin{cases} x_t & 1 \leq t \leq i \\ \max\{x_t, f_x(t-i, t-1)\} & i < t \leq k \end{cases}$$

and

$$\beta_i(t) = \begin{cases} f_x(t, t+i-2) & 1 \leq t \leq k+1-i \\ f_x(t, k) & k+1-i < t \leq k \end{cases}$$

It is very easy to verify that (6.4.a) reduces to (6.1) for i=k. Hence, our next step will be to check that (6.4) does satisfy (6.3). For i=1, (6.4.a) reduces to

$$\pi_1 = \Omega^{k-1} \ominus \alpha_1$$

where $T(\alpha_1)=k$, and

$$\alpha_1(t) = \begin{cases} x_t & t=1 \\ \max_{\delta}\{x_t, f_x(t-1,t-1)\} & 1<t\leq k \end{cases}$$

Since $f_x(j,j)$ is the maximum element in $\{x_1, x_2, \ldots, x_j\}$, it follows that $x_1=f_x(1,1)$ and $\max_{\delta}\{x_t, f_x(t-1,t-1)\}=f_x(t,t)$. Hence, we may write

$$\alpha_1(t) = f_x(t,t) \qquad 1\leq t\leq k$$

But from (6.4.b), we obtain for i=2

$$\sigma_2 = \Omega^{k} \ominus \beta_2$$

where $T(\beta_2) = k$ and $\beta_2(t) = f_x(t,t)$, $1\leq t\leq k$, which proves that $\beta_2 = \alpha_1$, and hence $\sigma_2 = \Omega\ \pi_1$.

The next step is to show that (6.4) does satisfy (6.3.b). For this, we sustitute (6.4) into the right hand side of (6.3.b) and denote the resulting sequence by ρ. This gives

$$\rho = \Omega \max_{\delta}(\Omega^{k-i} \ominus \alpha_i \ , \ \Omega^{k+i-2} \ominus \beta_i) \qquad 2 \leq i \leq k$$

Using property P2 to interchange $\Omega^{2(i-1)}$ and $\ominus$ in the second operand of $\max_{\delta}$ we obtain

$$\rho = \Omega^{k-(i-1)} \ominus \gamma_i \tag{6.5}$$

where $\gamma_i = \max_{\delta}\{\alpha_i \ , \ \Omega^{i-1} \beta_i\}$. By definition of $\max_{\delta}$, it follows that $T(\gamma_i) = T(\alpha_i) = k$, and

$$\gamma_i(t) = \begin{cases} \alpha_i(t) & 1 \leq t \leq i-1 \\ \\ \max\{\alpha_i(t), \beta_i(t-i+1)\} & i-1 < t \leq k \end{cases}$$

Hence with the definitions of $\alpha_i(t)$ and $\beta_i(t)$ we obtain

$$\gamma_i(t) = \begin{cases} x_t & 1 \leq t \leq i-1 \\ \max\{x_t \ , \ f_x(t-i+1,t-1)\} & t = i \\ \max\{\max\{x_t \ , \ f_x(t-i,t-1)\} \ , \ f_x(t-i+1,t-1)\} & i < t \leq k \end{cases}$$

Because of $\max\{\ \max\{a,b\} \ , \ c\} = \max\{a,b,c\}$, and $f_x(t-i,t-1) < f_x(t-i+1,t-1)$, we may rewrite γ_i as

$$\gamma_i(t) = \begin{cases} x_t & 1 \leq t \leq i-1 \\ \\ \max\{x_t \ , \ f_x(t-(i-1),t-1)\} & i-1 < t \leq k \end{cases}$$

from which we find that $\gamma_i(t) = \alpha_{i-1}(t)$, and hence, by

(6.5) and (6.4.a), that $\rho = \pi_{i-1}$. This proves that (6.3.b) is satisfied for the values of σ_i and π_i given by (6.4).

Finally, to check that (6.4) does satisfy (6.3.c), we substitute (6.4) into (6.3.c) and denote the resulting sequence by τ. This gives

$$\tau = \Omega \min_{\delta}\{\Omega^{k-i} \ominus \alpha_i \ , \ \Omega^{k+i-2} \ominus \beta_i\} \qquad 2 \leq i \leq k$$

$$= \Omega^{k-i+1} \ominus \min_{\delta}\{\alpha_i \ , \ \Omega^{i-1} \beta_i\}$$

In view of

$$\min_{\delta}\{\alpha_i \ , \ \Omega^{i-1} \beta_i\} = \Omega^{i-1} \varphi_i$$

where $T(\varphi_i) = T(\beta_i) = k$ and

$$\varphi_i(t) = \begin{cases} \min\{\alpha_i(t+i-1), \beta_i(t)\} & 1 \leq t \leq k-(i-1) \\ \beta_i(t) & k-(i-1) < t \leq k \end{cases}$$

we write

$$\tau = \Omega^{k+(i+1)-2} \ominus \varphi_i \qquad (6.6)$$

From (6.6) and (6.3.c), it follows that $\tau = \sigma_{i+1}$ only if $\varphi_i = \beta_{i+1}$. To prove this, we substitute the definitions of $\alpha_i(t+i-1)$ and $\beta_i(t)$ into $\varphi_i(t)$ and obtain

$$\varphi_i(t) = \begin{cases} \min\{\max\{x_{t+i-1}\ ,\ f_x(t-1,t+i-2)\}\ ,\ f_x(t,t+i-2)\} & 1 \le t \le k-(i-1) \\ f_x(t,k) & k-(i-1) < t \le k \end{cases}$$

But from lemma 2 in the appendix, and the fact that $f_x(t,t+i-1) = f_x(t,k)$ for $t=k-i+1$, we may write $\varphi_i(t)$ as

$$\varphi_i(t) = \begin{cases} f_x(t,t+i-1) & 1 \le t \le k-i \\ f_x(t,k) & k-i < t \le k \end{cases}$$

It follows that $\varphi_i(t) = \beta_{i+1}(t)$ and therefore that $\tau = \sigma_{i+1}$. This completes the proof that the sequences π_i and σ_i of (6.4) indeed satisfy the system of equations (6.3).

Now that (6.4.b) is known to be a valid formula for the sequence σ_i, we can easily obtain the network output sequence σ_{k+1} by setting i=k+1. This gives

$$\sigma_{k+1} = \Omega^{2k-1} \ominus \beta_{k+1}$$

where $T(\ldots)_{:+1}) = k$ and $\beta_{k+1}(t) = f_x(t,k)$, $1 \le t \le k$ which is identical with the expected output sequence (6.2).

7. Concluding Remarks:

This work was meant to contribute to the area of systolic architectures in three different ways, namely, by providing a mathematical model for systolic networks, an unambigious description of its input and output data, and a technique for the verification of its operation.

The central concepts in the present model are those of data sequences and sequence operators. Although we only defined the few operators that were used in the examples, it should be clear that other sequence operators may be introduced to model other types of computational cells.

A further step in this area is to develop a more complete sequence algebra to provide a basis for a solvability theory of the resulting system of difference equations on sequences. More specifically, it would be desirable to determine under which conditions an explicit analytical solution for the system of difference equations can be obtained. For a given network, this might determine, the properties to be satisfied by the successor function μ and the node I/O operators in order to verify analytically the operation of the network. If a sufficiently flexible algebra of this type were available, our model might prove to be very powerful in the design of new systolic networks.

At this point, we note that even if we cannot solve the resulting system of equations analytically, we can still use

a numerical iterative procedure to solve it. This approach is very close to the simulation of systolic networks, but appears to be more general and systematic.

Finally, we note that throughout this paper we assumed the systolic network to operate synchronously. However, the same model and techniques can be used for asynchronous networks. The only difference is in the interpretation of the i^{th} element of a data sequence, which now has to denote the i^{th} data item that appeared on a communication link instead of the data item that appeared on that link at time t=i.

References

1. Mead C. A. and Conway L. A., Introduction to VLSI systems, Addison-Wesley, Reading Mass. (1980).

2. Leiserson C. E., "Systolic Priority Queues," Proc. Conf. VLSI: Architecture, Design, Fabrication, California Institute of Technology, pp.199-214 (Jan. 1979).

3. Grefenstette J., Automaton Networks and Parallel Rewriting Systems, Ph.D. Dissertation, Dept. of Computer Science, University of Pittsburgh, 1980.

4. Misra J. and Chandi K. M., "Proofs of networks of processes," IEEE Trans. on Software Engineering, pp.417-426 (July 1981).

5. Von-Neumann J., Theory of self reproducing automata, University of Illinois Press (1966).

6. Ossefort M., "Correctness Proofs of Communicating Processes - Three Illustrative Examples from the Literature," TR-LCS-8201 (Jan. 1982). Department of Computer Science, University of Texas at Austin

7. Faurre P. and Depeyrot M., Elements of system theory, North Holland Publishers (1977).

8. Kung H. T. and Leiserson C. E., "Systolic Arrays for VLSI," Sparce Matrix Proc. 1978, pp.256-282, Society for Industrial and Applied Mathematics (1979).

9. Kung H. T., Class notes, Fall 1981.

10. Kung H. T., "Why Systolic Architecture," Computer Magazine, pp.37-46 (Jan. 1982).

11. Burks A. W., Essays on cellular automata, University of Illinois Press (1970).

Appendix

In the first part of this appendix, we list some properties of sequence operators that have been used in the paper. The verification of these properties is straight forward from the definitions of the operators involved. In the second part of the appendix, we prove two lemmas; the first gives an analytical solution to a difference equation that appears frequently in the verification of networks containing multiply/add cells, while the second one proves an equality that was needed in section 6.

Let ξ, ζ and η_j $j=0,1,2,\ldots,k$ be sequences in $\bar{R}_0$, and $w \epsilon R$; then

Property P1: $\Omega^r \Omega^k \xi = \Omega^{r+k} \xi$

Property P2: $\Omega^{(r+1)k} \Theta^r \xi = \Theta^r \Omega^k \xi$

Property P3: $w \cdot [\Theta^k \xi] = \Theta^k [w \cdot \xi]$

$w \cdot [\Omega^r \xi] = \Omega^r [w \cdot \xi]$

Property P4: For any binary operator 'op' extended from R_0 to $\bar{R}_0$, we have

$$\Omega^k [\xi \text{ 'op' } \zeta] = \Omega^k \xi \text{ 'op' } \Omega^k \zeta$$

$$\Theta^r [\xi \text{ 'op' } \zeta] = \Theta^r \xi \text{ 'op' } \Theta^r \zeta$$

Property P5: If η_j j=1,2,...,k are such that $T(\eta_j)=n$, then

$$\sum_{j=1}^{k} \Omega^{j-1} \eta_j = \Omega^{k-1} \eta$$

where $T(\eta) = n-(k-1)$ and $\eta(t) = \sum_{j=1}^{k} \eta_j(t+k-j)$.

The next result uses the $\oplus$ of (2.1):

Property P6: Let the sequences η_j, j=0,1,...,k satisfy $T(\eta_j) = n-j$, then

$$\eta_0 \oplus \Omega\, \eta_1 \oplus \Omega^2\, \eta_2 \oplus \ldots \oplus \Omega^k\, \eta_k = \gamma$$

where $T(\gamma) = n$ and

$$\gamma(t) = \begin{cases} \sum_{j=0}^{t-1} \eta_j(t-j) & t=1,2,\ldots,k \\ \sum_{j=0}^{k} \eta_j(t-j) & t=k+1,k+2,\ldots,n \end{cases}$$

Property P7: Given $\xi, \zeta \in \bar{R}_0$, then

$$\zeta * \Omega^r \xi = \Omega^r \gamma$$

where γ is described by

$T(\gamma) = \min(T(\zeta)-r\ ,\ T(\xi))$ and $\gamma(t) = \zeta(t+r) * \xi(t)$.

Lemma 1: The difference equation

$$\sigma_{i+1} = \Omega\, \sigma_i + \Delta_i \qquad i=1,2,\ldots,k+1 \qquad (a.1)$$

has the solution

$$\sigma_r = \Omega^{r-1}\,\sigma_1 + \sum_{j=1}^{r-1} \Omega^{j-1}\,\Delta_{r-j} \qquad r=2,3,\ldots,k+1. \qquad (a.2)$$

Proof: The proof uses induction on i. Evidently, for i=1 in (a.1) we obtain

$$\sigma_2 = \Omega\,\sigma_1 + \Delta_1$$

which is identical to (a.2) for r=2. Hence assume that for any r=1,2,...,k, σ_r is given by (a.2), then from (a.1) it follows that

$$\begin{aligned}
\sigma_{r+1} &= \Omega\,\sigma_r + \Delta_r \\
&= \Omega\,[\Omega^{r-1}\,\sigma_1 + \sum_{j=1}^{r-1} \Omega^{j-1}\,\Delta_{r-j}] + \Delta_r \\
&= \Omega^r\,\sigma_1 + \sum_{j=1}^{r-1} \Omega^{j}\,\Delta_{r-j} + \Delta_r \\
&= \Omega^r\,\sigma_1 + \sum_{j=0}^{r-1} \Omega^{j}\,\Delta_{r-j} \\
&= \Omega^r\,\sigma_1 + \sum_{j=1}^{r} \Omega^{j-1}\,\Delta_{r+1-j}
\end{aligned}$$

which proves that σ_{r+1} is also given by (a.2).

Lemma 2: let f_x be the ranking function for the set $X=\{x_1, x_2,\ldots x_n\}$, as defined in section 6, then

$$\min\{\max\{x_k\ ,\ f_x(i-1,k-1)\}\ ,\ f_x(i,k-1)\} = f_x(i,k) \qquad (a.3)$$

Proof: Let $y_1, \ldots, y_{k-1}$ be the result of sorting $x_1, \ldots, x_{k-1}$ in ascending order, and $z_1, \ldots, z_k$ the corresponding

result for $x_1, \ldots, x_k$. Hence, $f_x(i-1,k-1)=y_{i-1}$, $f_x(i,k-1)=y_i$ and $f_x(i,k)=z_i$. Now consider the following cases:

1) If $x_k < y_{i-1} < y_i$ then the left side of (a.3) is

$$\min\{\max\{x_k , y_{i-1}\} , y_i\} = \min\{y_{i-1} , y_i\} = y_{i-1}$$

Since $z_1,\ldots,z_k$ are obtained from $y_1,\ldots,y_{k-1}$ by inserting x_k in some position before y_{i-1}, we immediately see that $y_{i-1} = z_i$.

2) If $y_{i-1} < x_k < y_i$, then the left side of (a.3) is

$$\min\{\max\{x_k , y_{i-1}\} , y_i\} = x_k$$

and in this case it is clear that $x_k = z_i$.

3) If $y_{i-1} < y_i < x_k$, then the left side of (a.3) is equal to y_i, which in turn is equal to z_i because, in this case, x_k is inserted in some position after y_i.

Encl (1)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ICMA-82-47	2. GOVT ACCESSION NO.: AD-A120 698
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): A Mathematical Model for the Verification of Systolic Networks	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Rami G. Melhem and Werner C. Rheinboldt	8. CONTRACT OR GRANT NUMBER(s): Contract No. N00014-80-C-0455; Grant No. 80-0176
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Institute for Computational Mathematics and Appl. University of Pittsburgh Pittsburgh, PA 15261	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: Institute for Computational Mathematics & Appl. University of Pittsburgh Pittsburgh, PA 15261	12. REPORT DATE: October 1982 13. NUMBER OF PAGES: 61 pages
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Institute for Computational Mathematics & Appl. University of Pittsburgh Pittsburgh, PA 15261	15. SECURITY CLASS. (of this report): Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Systolic networks, formalization, verification procedures.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents a mathematical model for systolic architectures for use in the verification of the operation of certain systolic networks. The I/O description of the global effect of the computations performed by the network are obtained by solving a particular system of difference equations. The verification technique is applied to four different systolic networks proposed in the literature.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)